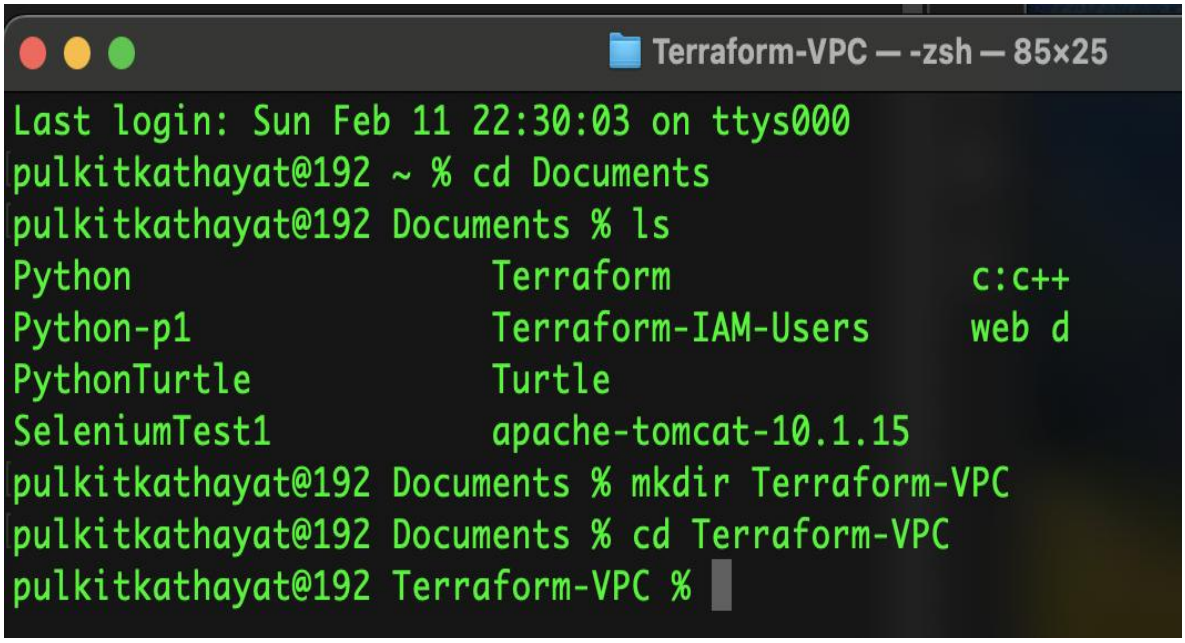
LAB-8 Creating a VPC in Terraform

Step 1: Create a Terraform-VPC Directory



Step 2: Create a main.tf

Welcome | main.tf

main.tf > resource "aws_vpc" "my_vpc" > tags > Name

```
terraform {
  required_providers {
    aws = {
        source = "hashicorp/aws"
        version = "5.35.0"
    }
  }
}

provider "aws" {
    region = "ap-south-1"
    access_key = "AKIATJHVFEM7OWRV3DM7"
    secret_key = "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"
}

resource "aws_vpc" "my_vpc" {
    cidr_block = "10.0.0.0/16"
    enable_dns_support = true
    enable_dns_hostnames = true

    tags = {
      Name = "MyVPC"
    }

}

resource "aws_subnet" "my_subnet" {
    count = 2
    vpc_id = aws_vpc.my_vpc.id
    cidr_block = "10.0.${count.index + 1}.0/24"
    availability_zone = "ap-south-1a"
    map_public_ip_on_launch = true

    tags = {
      Name = "MySubnet.${count.index +1}"
    }

}
```

Step 3: Initialize and Plan

```
Terraform-VPC — -zsh — 85×25
pulkitkathayat@192 Terraform-VPC % terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.35.0"...
- Installing hashicorp/aws v5.35.0...
- Installed hashicorp/aws v5.35.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
pulkitkathayat@192 Terraform-VPC %
```

```
pulkitkathayat@192 Terraform-VPC % terraform plan

Terraform used the selected providers to generate the following execution plan.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_subnet.my_subnet[0] will be created
  + resource "aws_subnet" "my_subnet" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "ap-south-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.1.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = true
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "MySubnet.1"
        }
      + tags_all                                       = {
          + "Name" = "MySubnet.1"
        }
      + vpc_id                                         = (known after apply)
    }

  # aws_subnet.my_subnet[1] will be created
  + resource "aws_subnet" "my_subnet" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "ap-south-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.2.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = true
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "MySubnet.2"
        }
      + tags_all                                       = {
```

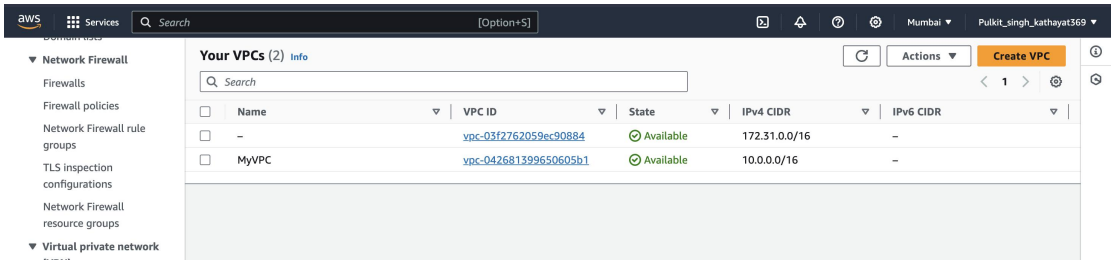
```
      + availability_zone_id                          = (known after apply)
      + cidr_block                                    = "10.0.2.0/24"
      + enable_dns64                                  = false
      + enable_resource_name_dns_a_record_on_launch   = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                            = (known after apply)
      + ipv6_cidr_block_association_id                = (known after apply)
      + ipv6_native                                   = false
      + map_public_ip_on_launch                       = true
      + owner_id                                      = (known after apply)
      + private_dns_hostname_type_on_launch           = (known after apply)
      + tags                                          = {
          + "Name" = "MySubnet.2"
        }
      + tags_all                                      = {
          + "Name" = "MySubnet.2"
        }
      + vpc_id                                        = (known after apply)
    }

  # aws_vpc.my_vpc will be created
  + resource "aws_vpc" "my_vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = true
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags                                 = {
          + "Name" = "MyVPC"
        }
      + tags_all                             = {
          + "Name" = "MyVPC"
        }
    }

Plan: 3 to add, 0 to change, 0 to destroy.

──────────────────────────────────────────────────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee
to take exactly these actions if you run "terraform apply" now.
pulkitkathayat@192 Terraform-VPC %
```

Step 4: Apply terraform

```
pulkitkathayat@192 Terraform-VPC % terraform apply

Terraform used the selected providers to generate the following execution plan. Resource
actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_subnet.my_subnet[0] will be created
  + resource "aws_subnet" "my_subnet" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "ap-south-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.1.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = true
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "MySubnet.1"
        }
      + tags_all                                       = {
          + "Name" = "MySubnet.1"
        }
      + vpc_id                                         = (known after apply)
    }

  # aws_subnet.my_subnet[1] will be created
  + resource "aws_subnet" "my_subnet" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "ap-south-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.2.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = true
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "MySubnet.2"
        }
      + tags_all                                       = {
```

```
          + "Name" = "MyVPC"
        }
      + tags_all                                 = {
          + "Name" = "MyVPC"
        }
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_vpc.my_vpc: Creating...
aws_vpc.my_vpc: Still creating... [10s elapsed]
aws_vpc.my_vpc: Creation complete after 11s [id=vpc-042681399650605b1]
aws_subnet.my_subnet[0]: Creating...
aws_subnet.my_subnet[1]: Creating...
aws_subnet.my_subnet[1]: Still creating... [10s elapsed]
aws_subnet.my_subnet[0]: Still creating... [10s elapsed]
aws_subnet.my_subnet[0]: Creation complete after 11s [id=subnet-0e159a1fcc50119e4]
aws_subnet.my_subnet[1]: Creation complete after 11s [id=subnet-08d2d58aa225245e9]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
pulkitkathayat@192 Terraform-VPC %
```

Step 5: Verify recources in AWS Console



Step 7: Clean Up

```
pulkitkathayat@192 Terraform-VPC % terraform destroy
aws_vpc.my_vpc: Refreshing state... [id=vpc-042681399650605b1]
aws_subnet.my_subnet[0]: Refreshing state... [id=subnet-0e159a1fcc50119e4]
aws_subnet.my_subnet[1]: Refreshing state... [id=subnet-08d2d58aa225245e9]

Terraform used the selected providers to generate the following execution plan. Resource
actions are indicated with the following symbols:
  - destroy

Terraform will perform the following actions:

  # aws_subnet.my_subnet[0] will be destroyed
  - resource "aws_subnet" "my_subnet" {
      - arn                                            = "arn:aws:ec2:ap-south-1:225999921982:subnet/subnet-0e159a1fcc50119e4" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "ap-south-1a" -> null
      - availability_zone_id                           = "aps1-az1" -> null
      - cidr_block                                     = "10.0.1.0/24" -> null
      - enable_dns64                                   = false -> null
      - enable_lni_at_device_index                     = 0 -> null
      - enable_resource_name_dns_a_record_on_launch    = false -> null
      - enable_resource_name_dns_aaaa_record_on_launch = false -> null
      - id                                             = "subnet-0e159a1fcc50119e4" -> null
      - ipv6_native                                    = false -> null
      - map_customer_owned_ip_on_launch                = false -> null
      - map_public_ip_on_launch                        = true -> null
      - owner_id                                       = "225999921982" -> null
      - private_dns_hostname_type_on_launch            = "ip-name" -> null
      - tags                                           = {
          - "Name" = "MySubnet.1"
        } -> null
      - tags_all                                       = {
          - "Name" = "MySubnet.1"
        } -> null
      - vpc_id                                         = "vpc-042681399650605b1" -> null
    }

  # aws_subnet.my_subnet[1] will be destroyed
  - resource "aws_subnet" "my_subnet" {
      - arn                                            = "arn:aws:ec2:ap-south-1:225999921982:subnet/subnet-08d2d58aa225245e9" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "ap-south-1a" -> null
      - availability_zone_id                           = "aps1-az1" -> null
      - cidr_block                                     = "10.0.2.0/24" -> null
      - enable_dns64                                   = false -> null
      - enable_lni_at_device_index                     = 0 -> null
      - enable_resource_name_dns_a_record_on_launch    = false -> null
      - enable_resource_name_dns_aaaa_record_on_launch = false -> null
      - id                                             = "subnet-08d2d58aa225245e9" -> null
      - ipv6_native                                    = false -> null
      - map_customer_owned_ip_on_launch                = false -> null
      - map_public_ip_on_launch                        = true -> null
      - owner_id                                       = "225999921982" -> null
```

```
  # aws_vpc.my_vpc will be destroyed
  - resource "aws_vpc" "my_vpc" {
      - arn                                  = "arn:aws:ec2:ap-south-1:225999921982:vpc/vpc-042681399650605b1" -> null
      - assign_generated_ipv6_cidr_block     = false -> null
      - cidr_block                           = "10.0.0.0/16" -> null
      - default_network_acl_id               = "acl-0b5474d640cafb860" -> null
      - default_route_table_id               = "rtb-090fc3c516872f10f" -> null
      - default_security_group_id            = "sg-099ce701b63a289b4" -> null
      - dhcp_options_id                      = "dopt-040306f93599488fe" -> null
      - enable_dns_hostnames                 = true -> null
      - enable_dns_support                   = true -> null
      - enable_network_address_usage_metrics = false -> null
      - id                                   = "vpc-042681399650605b1" -> null
      - instance_tenancy                     = "default" -> null
      - ipv6_netmask_length                  = 0 -> null
      - main_route_table_id                  = "rtb-090fc3c516872f10f" -> null
      - owner_id                             = "225999921982" -> null
      - tags                                 = {
          - "Name" = "MyVPC"
        } -> null
      - tags_all                             = {
          - "Name" = "MyVPC"
        } -> null
    }

Plan: 0 to add, 0 to change, 3 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_subnet.my_subnet[0]: Destroying... [id=subnet-0e159a1fcc50119e4]
aws_subnet.my_subnet[1]: Destroying... [id=subnet-08d2d58aa225245e9]
aws_subnet.my_subnet[1]: Destruction complete after 0s
aws_subnet.my_subnet[0]: Destruction complete after 0s
aws_vpc.my_vpc: Destroying... [id=vpc-042681399650605b1]
aws_vpc.my_vpc: Destruction complete after 1s

Destroy complete! Resources: 3 destroyed.
pulkitkathayat@192 Terraform-VPC %
```